AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph beginning at page 5, line 19, with the following:

A1

By storing the mapping table 200 in volatile memory, the table 200 can be easily and rapidly accessed and modified on the agents 110. Storing the mapping table 200 in volatile memory has the further advantage of substantially reducing the cost and complexity of implementing the agents 110 as mapping agents. Overall, the agents 110 allow the performance-sensitive mapping process to be parallelized and distributed optimally for performance. The mapping agents 110 ~~that~~ may reside on a host 140 ~~are~~ or elsewhere on the Network. ~~The agents 110 combine with the controller 120 to fabricate the existence of the virtual disk drive 150~~.

Please replace the paragraph beginning at page 7, line 14, with the following:

A2

This disclosure describes the mapping table 200 as having one entry 210 per each "disk block" of virtual disk 220. While possible to build, this would result in huge mapping tables and highly fragmented mapping, both of which introduce undesirable performance degradations. In another implementation, each mapping table entry 210 represents a variable sized group of contiguous virtual disk blocks that map to contiguous blocks on one of the physical storage devices. This configuration of the mapping table 200 offers mapping flexibility and dense mapping structures, but introduces greater algorithmic complexity in managing the variable sized blocks and greater map entry lookup costs. Therefore, the table 200 may use mapping table entries ~~220~~ 210, each having a fixed size number of contiguous blocks ("segments") on the virtual disk 150 that map to one storage device. While this

configuration for the table 200 is possibly not as dense as variable sized block mapping, the configuration offers the simplest and highest performance map access and space management. In this configuration, each of the entries 210 contains a virtual disk segment 220 instead of a virtual disk block. Regardless of the specifics of the table 200, the table 200 must map a virtual drive segment 220 to each physical storage block involved in I/O operations.

Please replace the paragraph beginning at page 8, line 4, with the following:

In another configuration, the system 100 has multiple tables 200, each having different mappings between a virtual disk 150 and the storage devices. In this way, different hosts 140 may have different access paths to the same storage device. When the mapping table 200 does not include one of the storage locations 230, hosts 140 using this table (i.e., the hosts 140 connect to the agent 110 that stores this table) cannot access information stored at a storage location. In fact, the host 140 will not even realize that this storage location 230 exists.

Please replace the paragraph beginning at page 9, line ~~28~~ 26, with the following:

As presented above, the Nw state 250, when activated, indicates that any write operations to the virtual disk segment(s) 220 represented by the entry 210 cause the agent 110 to send a fault message the controller 120. The agent 110 does not allow the host 140 to write to the storage locations 230 until the controller 120 returns a fault response to deactivate the Nw state 250. Unlike the invalid state 240, the activated Nw state 250 does not prevent read operations from generating faults. Instead, the agent 110 generally allows the host 140 to proceed to access data at the storage location 230. Accordingly, if only the Nw state is activated, the mapping table entry 210 must

contain a useable storage location 230. ~~It should be noted that this invention covers alternative methods of allowing the write to proceed, for example, the do_write command would allow a write to proceed despite an I/O fault.~~

Please replace the paragraph beginning at page 10, line ~~25~~ 23, with the following:

In another configuration, the mapping table 200 further includes an error (E) state 270. When active, the E state 270 indicates the existence of an error condition and provides the information necessary to instruct the agent to return an error without disrupting any previous state. The E state 270 is used where a pre-existing failure is known and such failure would cause any attempts at I/O access to fail. It should be noted, however, that the E state 270 could also be used as the means to issue an error status from a mapping fault. If an entry 210 contains an active E state 270, the agent 110 ignores the storage address 230. If the host 140 attempts to read from or write to the storage address 230, the agent 110 returns an error to the host 140.

Please replace the paragraph beginning at page 17, line ~~23~~ 21, with the following:

The controller 120 then updates its persistent copy of the mapping table 200 for all the virtual disks 150 that share the faulting segment ~~230~~, except for the mapping table that maps the particular virtual disk 150 associated with the I/O fault, step 550. In particular, the controller 120 remaps the virtual disk segments 220 to the newly allocated storage location 230. To update the mapping tables, the controller 120 deactivates the Nw state 250 in the persistently stored table. As part of the step

550, the controller 120 changes the storage location 230 to refer to the newly allocated segment.

Please replace the paragraph beginning at page 18, line ~~3~~, with the following:

In step 560, the control 120 sends set_entry commands to all mapping agents 110 that ~~have write faults and remain on the old storage location~~ use the updated mapping table. This action propagates the segment change and the Nw state change to these mapping agents 110. The set_entry activates the blocking flag 280, allowing the controller 120 to know when all outstanding read I/Os to this segment have finished before allowing any writes to proceed to the original segment. The controller 120 waits for these set_entry operations to complete before acting further.